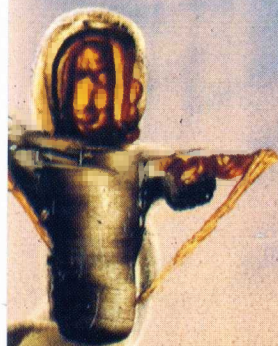


Watson's
Notes

SPECTRUM +

COMPUTER GAMES



D. KEDEM & I. KALISKY

SPECTRUM+

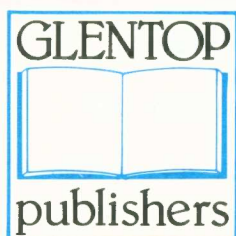
*Watson's
Notes*

SPECTRUM+

I.KALISKY & D.KEDEM

UNIT 3

Computer Games



NOVEMBER 1985

All programs in this book have been written expressly to illustrate specific teaching points. They are not warranted as being suitable for any particular application. Every care has been taken in the writing and presentation of this book but no responsibility is assumed by the author or publishers for any errors or omissions contained herein.

COPYRIGHT © Glentop Publishers Ltd 1985

Translated from the original.

COPYRIGHT © I.Kalisky & D.Kedem 1984, 1985

No part of this publication may be copied, transmitted or stored in a retrieval system or reproduced in any way including but not limited to photography, photocopy, magnetic or other recording means, without prior permission from the publishers, with the exception of material entered and executed on a computer system for the reader's own use.

ISBN 0 907792 62 6

Published by:

Glentop Publishers Ltd
Standfast House
Bath Place
High Street
Barnet
Herts EN5 1ED

Tel: 01-441-4130

Cover illustration from an original painting by Nick Woods

CONTENTS

	PAGE
Foreword	6
Chapter 1 - Shooting the Aliens	7
Chapter 2 - Video Pencil	19
Chapter 3 - Horseracing	31
Chapter 4 - Treasure Hunt	43
Summary Index	46
Answers	47

FOREWORD

You have now reached the third unit of the course. You have already learnt a considerable amount, which enables you to write programs that can produce interesting graphics, while using the different colours and sounds generated by the computer. We also hope that you feel you are beginning to gain control over the computer.

We have devoted the third unit to the development of computer games. During the development of these games you will learn more BASIC commands and the concepts connected with program writing.

By the end of this study unit you will have sufficient tools to develop many interesting programs of your own.

Dani Kedem and Itzhak Kalisky

Chapter 1

SHOOTING THE ALIENS



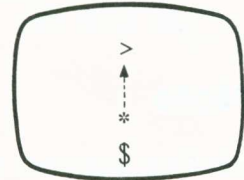
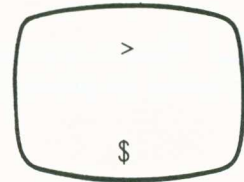
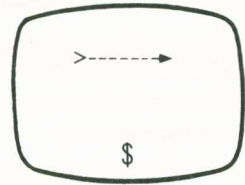
Who among us hasn't played, if only once, with one of those thousands of video games?

This time, for a change, you are going to develop such a game yourself.

1. The Aliens' spaceship, a '>' crosses the screen repeatedly from left to right. Your spaceship (\$) is preparing an ambush at the bottom of the screen.

2. You press a key at the moment you think the aliens ship is located exactly above your spaceship.

3. At this point the alien's spaceship halts and the missile that you shot '*' begins to climb upwards from your space ship.



The aim of the game is to hit the alien's ship '>'!

Let's check what you know and what you need to learn to develop the game.

- Do you know how to move the aliens repeatedly from the left side of the screen to the right side ____ (YES or NO)?

? Why does this happen?

? What is the 'secret' of this short program?

The answer is in the INKEY\$ command.

When the computer reaches the INKEY\$ command in the program, it prints out the key that is being pressed at that moment and then carries on with the execution of the rest of the program.

? What happens if no key is being pressed at that moment?

In that case the computer ignores the INKEY\$ command and carries on. If this is so, what does the computer do when it reaches line 10?

The program works as follows:

```
10 PRINT INKEY$;
```

This prints the character of the key being pressed and moves to the next line:

```
20 GO TO 10
```

The program then loops back to line 10.

? When you do not press a key, does the computer stop and wait at line 10, or does it continue to RUN through the program in an indefinite loop?

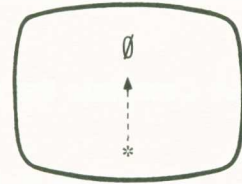
Answer 1

? Why does the program print the same character several times, if you continue to press the key?

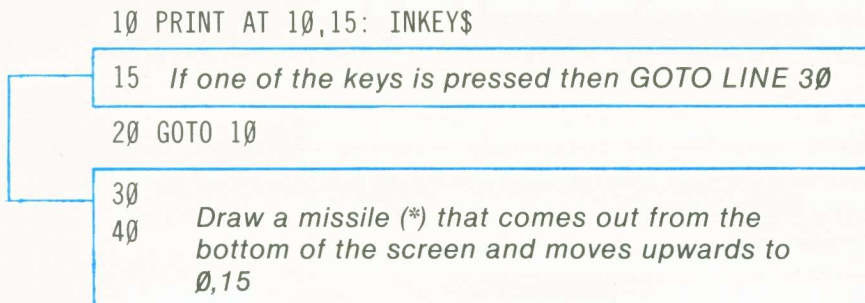
Answer 2

? Would you like to convince yourself that the computer actually runs about in a loop even when you don't press any keys?

If you press \emptyset (the trigger), then a 'missile' (*) will be launched from the bottom of the screen, hitting the letter in the middle of the screen and disappear into the blue sky!



The program will look like this:



The program works as follows:

As long as you don't 'pull' the trigger (\emptyset), the computer ignores line 15, cycling between lines 10 and 20. It displays any character you press, in the middle of the screen.

As you press the \emptyset key the computer 'jumps' to line 30 and launches the missile up the screen.



How can you write line 15 in BASIC?



It is done like this:

```
15 IF INKEY$="0" THEN GOTO 30
```

This line means that **if** \emptyset is pressed (INKEY\$="0"), **then** goto line 30!

IF...THEN



And if \emptyset is not pressed?

In that case the computer simply goes on, to the next program line (20).

- Complete the program and run it.
- It would be better if you add a BEEP command between lines 10 and 20. This will convince you that the computer cycles between these lines, even when a key is not pressed.
- You should add commands, that will draw a missile that flies up the screen, from line 30 onwards.
- Check the program, press different keys, as well as the 0 key.

Answer 5

Note the 0 must be between quotation marks.

? Why?

You will understand that later in this study unit.

- Improve the program so that it will not end after you shoot your first missile, it would be better if you were able to shoot a new missile each time you press the 0 key.

Answer 6

? What will happen if the following line is added to the program?

```
17 IF 2=3 THEN PRINT "KATE"
```

Answer 7

Now, you know everything you need to know in order to write the program described at the beginning of this unit.

But, since the IF...THEN command is of such great importance in the computer language, let us practice it a little further:

? What do you think will happen if you add the following line to the program?

```
18 IF INKEY$="Z" THEN NEW
```

Add line 18 to the program. RUN the program. Shoot the missiles and at the end press the 'Z' key!

? Are you surprised at the result?

If you do not understand what happened then refer to the answer.

Answer 8

? Do you remember what is carried out by the following program:

```
10 FOR N=3 TO 10
20 PRINT AT N,N;N
30 NEXT N
40 PRINT "END"
```

If you don't, type the program in and RUN it.

Write a program that does exactly the same thing, now without using the FOR...NEXT command!

Clue

Use IF...THEN (LET and GOTO will also help you).

RUN your program and check it before you compare it to the answer.

Answer 9

Note

Since the IF...THEN command is a condition, it is frequently used in conjunction with conditional symbols such as '>', '<' and '<>', i.e. 'greater than', 'less than' and 'not equal to'. These symbols are used to define the condition. For example, with X as a variable:

```
IF X > 10 THEN...
IF X < 10 THEN...
IF X <> 10 THEN...
```

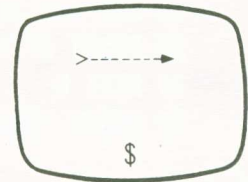
Now it is time to go back to our game.

We shall remind you of what the program should do:

1. The aliens space ship (>) crosses the screen repeatedly from the left to the right.

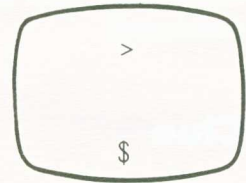
```
FOR X=.....NEXT X
```

Your space ship (\$) is preparing an ambush at the bottom of the screen, (place your space ship at coordinates (21,15)).



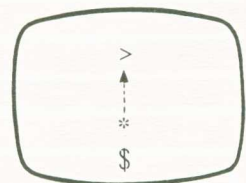
2. At the moment you think that the 'aliens' are located exactly above your space ship, you press any of the keys that you had previously chosen to act as the 'trigger'.

```
IF INKEY$=...THEN GOTO
```



3. At the moment the aliens space ship halts, the missile you shoot (*) should begin to climb directly upwards!

```
FOR Y=...NEXT Y
```



The aim is to hit the alien's space ship!

With the help of the clues given above and with the knowledge you have acquired you can proceed with the development of the program.

Build the program in stages: write the first stage and then RUN it. Afterwards add the second stage, RUN it and so on.

Find the 'bugs' in your program and solve them.

Note

It is possible that the program you write will be different from the one we suggest as a solution, that should not cause any problems. Compare the two programs and see which one is shorter. Usually more than one program can be written to carry out the same task (as you can see on page 13).

After writing the program and running it, it is possible you have found the following 'bugs':

- The missile wipes out your own ship, therefore you should start firing it from higher up.
- On pressing the trigger the alien ship disappears, therefore you should transfer the command 'IF INKEY\$...' to a more strategic position in the program .

If you have not already done so, enter these improvements into the program so that the game will continue after the first missile has disappeared.

Possible Bugs

An enemy ship that wasn't hit (by a shot previously made) stays on the screen and interferes, the use of CLS in the appropriate place will solve this.

The full program is in the back of the book.

Answer 10

By the way, you will get more satisfaction from solving (or at least trying to solve) the problems yourself and then looking at the answers rather than looking at them straight away.

Perfecting the program

Diligent programmers will always try to improve and perfect the programs they write!

Let's improve the program.

To remind you: the Spectrum+ computer is rich in colours and sounds, take advantage of this for your own enjoyment!

 Enter commands to tell the computer to display on the screen, after firing each missile, the total number of missiles you have fired.

Clues:

- You will need a 'count' (such as $N=N+1$), remember the digital watch!
- Put the count in a suitable place in the program. RUN the program and check that it is counting the missiles you are firing.
- Don't forget to set the count to zero at the start of the program.

- Display the count in a place on the screen that won't interfere with the game.

Answer 11

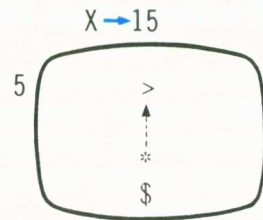
Another Refinement

? Are you fed up with counting in your head the number of your hits you have made on the target?

The computer will happily do this for you if you tell it too!

Let's see what this means in computer terms.

If you think about it and then answer the following question, you will know how to tell the computer to count your hits on the target.



? You've fired a missile, and it rises to the top. In which column (X coordinate) must the enemy ship be in order for the missile to hit it — (15, 10 or 5)

(If you don't remember what a column is refer to Unit 2.)

? Let us assume that you have chosen (like us) variable M as a count for hits on the target. Which of the following commands will suit you:


- a. LET M=M+1
 - b. IF X=15 THEN LET M=M+1
- or c. IF Y=15 THEN LET M=M+1

Answer 12

It only remains for you to set the hit count to zero, to enter the command 'IF' in the correct place, and tell the computer to display M in a convenient place on the screen.


Make the improvements to the program and check if it functions to your satisfaction.

Answer 13

 Tell the computer to **stop** running the program after firing 10 missiles.

STOP
Answer 14



 Tell the computer to print out the **hit percentage** before it stops.

Answer 15


Ending the Game

When you wish to end this game don't just type **NEW**: insert a change so that if you hit the enemy ship 10 times out of 10 attempts, the whole screen will 'destruct' and the program disappear!

Chapter 2

THE VIDEO PENCIL: DRAWING WHAT YOU WANT

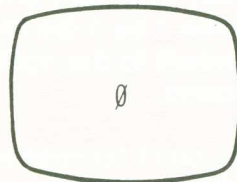
You are ready to construct a program that will turn the computer into a 'video pencil' and, by pressing the appropriate keys, you can draw what you wish on your television screen.

 Erase everything from memory (using NEW).

Let us define exactly what the program needs to perform.

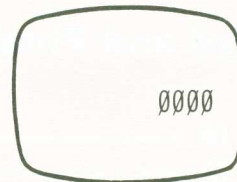
1. Having pressed RUN the number 0 should be displayed in the centre of the screen.

RUN:



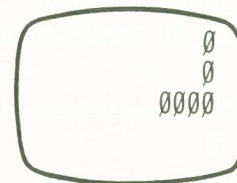
2. On pressing → the number 0 moves to the right, trailing a line of 0's

→ :



3. By pressing ↑ the trail of 0's moves upwards. The 0's are drawn in the direction of the arrow pressed while you continue to hold down the key. This is true for the directions of all keys: up, down, left and right, (↑, ↓, ← and →).

↑ :



Note: we have chosen the arrow keys for convenience. In the same way we could have chosen any 4 other keys! We will call them the pencil keys.

You have all the knowledge needed to write a program like this. Try and write it without going on to the clues.

First Clue:

The program is built around two variables X and Y, in the command:

```
PRINT AT Y,X;Ø
```

When you press one of the pencil keys, X and Y will change their value depending on the key pressed. The computer will display the Ø in the appropriate place, depending on the values of X and Y. (This is with the help of the IF...THEN commands.)

Second Clue:

If the key pressed (using INKEY\$) is CHR\$ 9 (the key with the ↵ drawn on it), then increase X by 1 (as this key indicates movement to the right) and return to the PRINT command.

(If you don't remember how to use CHR\$, see Unit 2, page 34.)

Third and Final Clue:

```

1Ø _____
2Ø _____
3Ø PRINT AT Y,X;Ø
4Ø IF INKEY$=CHR$ 9 THEN LET X=X+1
5Ø _____
6Ø _____
7Ø _____
1ØØ _____

```

} the 4 keys
X=X+1
with the
arrows

Answer 16

Try to move the video pencil off the edge of the screen.

What happens?

If you try to leave the screen on the right or bottom, the program will stop. If you try to leave the screen to the left or top the video pencil will automatically change its direction! Later you will overcome this problem.

Perfecting the Program

Change the program so that the tip of the pencil will be the number 2 instead of 0.

We will now enter an improvement in the program.

Before starting to draw the computer will ask you which character will be the tip of the pencil.

For this you must learn a new command:

INPUT

First of all, you must change the number 2 in the PRINT command in your program, to a variable:

30 PRINT AT Y,X;A



Wait!

Do not run the program yet. The computer doesn't know the value of A! (if you RUN it, the computer displays error message number 2.)

A line before the PRINT command enter the instruction:

INPUT

INPUT A



which means: Enter a value into variable A.

Answer 17

Run the program.

appears in the bottom left hand corner of the screen.

The computer has reached the command INPUT A, has stopped and is waiting for you to type in the number you want A to be.

Choose a number type it in and press ENTER.

Start drawing beginning at the number on the screen.

Do you want to change pencils?


Stop the program running (BREAK), RUN it again and enter the new number.

To Conclude

Every time the computer reaches the instruction INPUT A in the program, it stops and signals as it is waiting for you tell it what to put in A, in the INPUT statement.

Another Improvement


Change the 'pencil tip' (changing the number in A) without erasing the existing drawing!

 Enter an extra instruction in the program that will do the above: i.e.

'If key 'P' (for example) is pressed, go to the INPUT instruction.

Answer 18

 Run the program.

Each time you want to change pencils, press P. The  sign will appear in the bottom left corner, to tell you that the program is waiting an INPUT. Type in a number and press ENTER.

Now you can continue to draw with your new pencil.

A Musical Pencil...

Task

Enter a line into the program such that when X or Y are changed a musical tone is made, which will increase in pitch when X or Y increase, and will decrease in pitch when X or Y decreases in size.

Note: when you do not press the keys the musical tune must not be made.

Answer 19

An Improvement

If you are fed up of drawing with only a number, you may wish to use any of the keys, i.e. numbers, letters and even graphics characters!

When the program waits at the INPUT, try to enter a letter instead of a number.

The computer will refuse to accept it and will 'throw it out' from the program, giving the error message 2.

? Why is it impossible at this stage to enter, in the INPUT statement, any letter of the alphabet?

When the computer stops due to the INPUT A instruction.

? It waits for you to enter the value of _____ (Ø A or P)?

Variable A can only be a number, and not an alphabetic or graphics character (A is a numeric variable). Therefore the computer would not accept the letter, because it was expecting a number.



String Variables

The improvement you will make in the program will assist you in overcoming this problem.

A appears in the next 2 lines, add the \$ immediately following the A:

```
INPUT A$  
PRINT ----- A$
```

Run the program.

In the bottom left corner of the screen an interesting thing appeared: the  sign appeared in quotation marks "".

Type in any character you think of, graphic characters. (If you don't remember how to get these, see page 5Ø, Unit 1), numbers, letters, or even words and sentences! (Do not forget to press enter however.)

Now you can draw, these different characters will be your pencils (to change pencils press 'P').



Have you finished drawing?

Let's explain what has happened.

A was a numeric variable.

From the following values circle those that could be held in A.

0.1,  , STOP, -1.4, C, 3

Answer 20

A\$ in comparison is a **string variable**.

String Variable



What is a string?

You have already used strings on past occasions without even knowing that they were strings.



What does the computer do with the following instruction:

```
PRINT ."2+5"
```

The computer does not calculate the sum, but simply displays whatever is written in the quotation marks.

Anything in the quotation marks is taken as a string.

STRING

```
"STRING"
```

As with a string of beads on a thread, elements of a string do not require logical connections with each other.

The computer will always copy a string parrot fashion.



In the following instruction.

```
PRINT "4*7=";4*7
```

which part is string and which part is arithmetic expression?

Answer 21

By the way:

Assume that the computer is waiting for you on the INPUT A\$ instruction and exactly at this moment you want to stop the program running.



Run the program, in the bottom left corner of your screen appeared, of course:

```
"L"
```




Now try to press BREAK.



What does the computer understand it to be ____ (BREAK or SPACE)?



How, then will you stop the program?

Move the  to the left and outside the quotation marks and press STOP.



Now on ENTER and the program will stop!

Let's return to strings

We have said that A\$ is a string variable.



Can you recognise the string variable in the following program:

```
500 LET B=10
510 LET B$="XØR"
320 PRINT AT B,B;B$
```

? What in your opinion will the program do!

Do not erase the pencil program!

Add this program to the pencil program.
Pay attention to the line numbers.

In order to RUN the 3 lines 500, 510 and 520 type:

```
RUN 500
```

and press ENTER.

This instruction tells the computer to start the program from 500 and to run from here on!

? Have you finished?

Now erase those three lines.

Summary

In the pencil program we wanted to draw with the assistance of any character we chose. Therefore in the following instruction:

```
PRINT AT Y,X;A$
```

we have used a string variable (A\$) which can accept any character or digit we want, or even a combination of numbers and characters together (string), and not just numbers as does the numeric variable A.

? What will happen if we change the line to:

```
PRINT Y,X; "A$"
```

Try it and see!

? Does the computer in this case treat A\$ as a string or as a string variable?

Note:

The name of a string variable must be one alphabetic letter!



Which of the following string variables are not valid?

A\$, BB\$, C3\$, 4\$, K\$

Answer 22

A Coloured Pencil

Let us return to the video pencil.



Change line 30 back to:

```
PRINT AT Y,X;A$
```



Make this change in the program, so that the computer will ask you in what **colour** you want to draw (and of course it will draw in this colour).

Colour

Answer 23

Improvements for the Advanced Programmer



Write another 4 instructions using the structure IF...THEN..., that will stop the pencil moving off the screen. Each time the pencil reaches the edge of the screen and you press an arrow key to continue moving in that direction off the screen, the pencil will refuse to fulfill your instruction and will remain at the edge of the screen.

Self Destructing Programs



Do not erase the program already in memory.

Type into the program an instruction that, when the pencil reaches the square 10,8 for instance, the program will destroy itself! (That is to say, you put a mine in the square, (10,8). If you then step on it the program will destroy itself (using NEW).

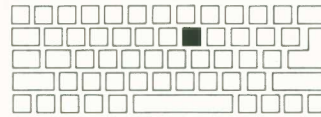
Therefore you have to learn a new command:

AND

If $Y=8$ **and** $X=10$ then destroy the program, i.e. in BASIC

AND

IF $Y=8$ AND $X=10$ THEN _____



Complete this instruction and enter it into the program. Try to walk on the mine.

Further Ideas

Possibly, this mine idea gave you an idea for a 'Minefield' game? For example: you have to reach the treasure at the bottom of the screen, from the top without stepping on the mines which you have scattered on the screen.



Chapter 3

HORSE RACING

The Computer as an Electronic Die

Let's start with a program that will make the computer act as an electronic die (that's the singular of dice).

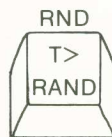
Pressing a key will 'Draw the side of a die' on the screen to show you which number has come up!

Type in and run the following program.

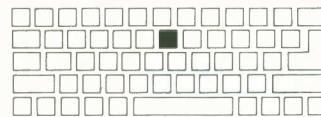
```
10 PRINT RND
20 GO TO 10
```

RND tells the computer to select a random number.

Highlight →



RND



Type:

```
PRINT RND
```

Note:

There is a difference between RND and RAND, make sure RND is what is displayed, you will have to go into Extended mode to achieve this.

Each time you execute scroll the computer will continue and show you more numbers.

(If you don't remember how, See page 43 Unit 1).

Try it.

So now we can produce our own RaNDom numbers, however, let us return to our die.

? Are all the numbers you have got greater than _____ ($0.1, 0, 0.5$) and lower than _____ ($1, 0.9, 2$)?

Whenever the computer displays $.06374$ for example, it means 0.06374 .

That is to say, the computer has chosen random numbers between 0 and 1 . RND will never pick the number 1.00000000 (one).

Change the program such that the computer, will choose numbers, between 0 and 6 .

```
10 PRINT RND* _____
20 GOTO 10
```

Answer 25

□ Try and see why you have got numbers between 0 and 6 .

INT, (short for INTeger) gives a whole number. In order to create a die, we get rid of the fractions following the decimal point.

INT

4.23076682

That's how we will get the six numbers of the die.

□ Change your program to the following:

```
10 PRINT INT (RND*6)
20 GO TO 10
```

Brackets



Integer means whole number

The action of INT is very simple: it takes from the number, the part to the left of the decimal point, only whole number.



In your opinion what is:

```
INT 7.99
```

equal to?



Think first and then type in:

```
PRINT INT 7.99
```

and press ENTER.



Run the program in the computer again and answer the following questions:



On the screen the number 0 will probably have appeared, can you find this number on a real die ____ (Yes or No)?



On a real die you can find the number 6. Did it appear on your electronic die ____ (Yes or No)?

That means that your electronic die selects numbers from 0 to 5, instead of numbers 1 to 6 as on a real die.



Change your program so that it will act as a real die:

```
10 PRINT INT (RND*6)+ ____
20 GOTO 10
```

Answer 26



Run your program again and check again if each of the numbers 1, 2, 3, 4, 5 and 6 appeared on the screen, at least once.



Write a program that will do the 'pools' for you:

```
0 = a draw
1 = a home team win
2 = away team win
```

Answer 27

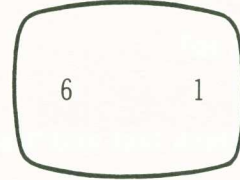
Now you can have some fun.

Write a program such that every time you press a key (for example L) the results of throwing two dice will appear on the screen.

Note: clear the screen (CLS) between each throw.

(Add colour so that every number will appear on a different PAPER: PAPER 6 for colour e.g red and so on.)

Change the program so that before every 'throw' of the dice the computer will make a sound, the volume of which will depend on the result shown on the dice.



Answer 28

A 'Celebration' on the Screen

Add one line, so that if the two dice show the same number then you will hear different sounds and the BORDER will change colour!

Clue: Use a colon(:) in the desired line.

Answer 29

What do you think the next program will do?

```
10 PRINT AT INT (RND*22),INT
  (RND*32);" ■"
20 GO TO 10
```

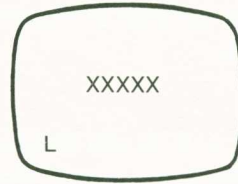
After you've thought enter it into the computer and run it.

Change the program so that every square will be in a different colour and will make a different sound when displayed.

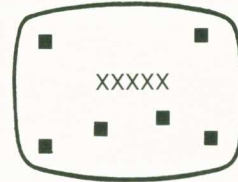
Answer 30

Game Corner, Battleships

1. The computer displays on the screen your submarine (xxxxx - or any other shape that you choose) and asks you (using INPUT) how many missiles to launch at the enemy submarine.



2. Lets say that you've chosen 40 missiles. The computer starts launching the missiles (■) onto the screen in any shape (using RND) until the whole screen is filled with the 40 missiles.



If your (xxxxx) submarine was not hit then you receive 40 points. If it was hit (x■xxx) you don't receive any points (or you lose 20).

You can play against a friend (each one goes in turn) or against the computer and it will choose the number of missiles it requires to sink your ships/submarines.

Answer 31

Disco Corner

? Have had a disco party in your house before?

See if you can get the computer to produce similar music and flashing lights.

□ Write a program that will change the BORDER colour with RND to about the same tempo (using PAUSE) that of the music.

Horse Racing

Let's go into the open air to the horse racing track.

1. 1, 2, 3 are three horses that are getting ready for a race from the left side to the right side of the screen, whoever gets to the right side first is the winner.

```
111
22222
33
```

2. The computer chooses a number from 1, 2 or 3. If, for example number 3 comes up, the computer will add to horse number 3 a 1.

```
111
22222
333
```

3. Horse number 1 has won!

It arrived first at the right side because it's number came up more times than the other horses.

```
111111111
2222222
33333333
```

Once again the knowledge you've gained so far is sufficient for you to write this program!

Write a program that will perform the horse race. If it is still difficult for you here is a clue.

A Clue:

First try to write a program of a race with only one horse. This horse will only proceed if the computer chooses the number 1 from the three numbers: 1, 2 and 3.

line 3
(for example)

```
X1
1111111
```

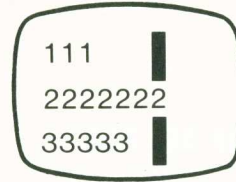
Answer 32

Now write the full program for the three horses!

Answer 33

Improvements

Draw on the screen a finishing line that the horse have to reach. In this example horse number 2 has won because it reached the finishing line.



Tell the computer to stop the race when a horse has won. To do this you have to use **OR**!

Answer 34

OR



If X1= _____ OR X2= _____ OR X3= _____ THEN STOP **Answer 35**

Add a command that will make the computer ask you before each race, on which horse you're gambling. (INPUT)

Answer 36

Shortening the Program Using DIM

The horse racing program included about 18 commands. You can drastically improve the program by adding more horses, yet shorten the program to half its length!

How?

For this you have to learn the command **DIM**. (DIM is an abbreviation of the word DIMension.)

DIM

Erase any previous program.

 Enter the following command:

```
10 DIM X(3)
```



(make sure you use the brackets)

This command did, all at once, the same as the three commands in the horse racing program:

```
10 LET X1=0
20 LET X2=0
30 LET X3=0
```

When the computer received the command:

```
DIM X(3)
```

it performed the following:

The computer set aside three variable locations:

```
X(1), X(2), X(3)
```

The computer gave to each one of them an initial value of 0.

This group of variables is called an **ARRAY**.

Array

They all have the same variable name, X.

Each of them have different numbers. For example:



 Run the following program:

```
10 DIM X(3)
```

Tell the computer to print the value of: X(1).

```
PRINT X(1)
```

and press ENTER

What did the computer print on the top of the screen ____ (1 or 0)?

In the same way ask the computer to print the value of the rest of the array, X(2) and X(3).

What will happen if you ask the computer about X(4)?

Do it!

The computer will display error message 3

You asked the computer to print the value of X(4) but you only had an array of three values DIM X(3). The computer did not set up a fourth place. It lets you know this in the error message.

Does the computer recognise X2 (without brackets)?

Try it!

You can see that the computer does not recognise this, as it gives the error message number 2.

What do you think will happen in this next program:

```
10 DIM A(4)
20 FOR N=1 TO 4
30 INPUT A(N)
40 NEXT N
```

Run the program.

The computer asks you for the value of A(1) the first time around the loop.

Choose any number and press ENTER, the computer will now ask you for the value of A(2) and so on.

Pay attention to the way we've used N, to give each one of the numbers in the array!

Tell the computer to print A(1).

? Is this the value you gave it?

Now add another three lines to the program so that after filling all parts of the array the computer will automatically display them on the screen, as follows:

A(1) = (The value you put in the first variable)
A(2) = (The value you put in the second variable)
A(3) = (The value you put in the third variable)
A(4) = (The value you put in the fourth variable)

? Are you finding it hard?

If so please refer to the answer.


Answer 37

After you have done this RUN the program.

Now erase the program and type in and RUN this second program.


```
1Ø DIM C(3)
2Ø LET N=INT(RND*3)+1
3Ø LET C(N)=5
4Ø PRINT N
```


? Which one from variables C(1), C(2) and C3 equal zero? and which of them equals 5?

 In order to convince yourself that you are right tell the computer to print their values $C(1)$, $C(2)$ and $C(3)$

Explanation

Let's assume that the result of the calculation in line 20 was 2. The computer arrived at line 30 and put 5 in $C(2)$. The other two variables remained at zero.

 Run the program again and you will realise that the computer gives one of the variables the value of 5 according to the result of the calculation. (Depending to the value of N .)

 What will the program do if we change the instructions to:

```
30 LET C(N)=C(N)+5
35 BEEP 0.1,6
40 GOTO 20
```

 Run the program.

 Can you see anything on the screen?

 Is the program continually running?

The fact that it is making these noises proves that it is.

Now the program awards the prize to the winner.

Let's assume that number 3 won, so it increases the value of $C(3)$ in line 30 by 5 and returns to allow the others to finish the race.

 Stop the program.

Do you think that the values of the 3 variables C(1),C(2) and C(3) are equal or different?



Tell the computer to show you the values of the 3 variables.



What will happen to the program if you add the line:

```
37 PRINT AT N*3,10;C(N)
```

After you have thought about it, type in the line and RUN the program.

Returning to 'Horse Racing'.

Now you have all you need to improve the horse racing, shorten the program with DIM, and increase the number of horses to 6.

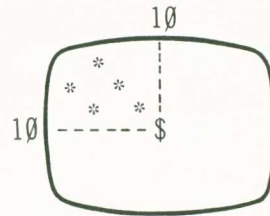
Answer 38

Chapter 4

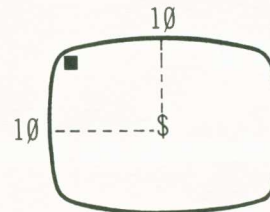
TREASURE HUNT: A FINAL GAME

In this game we are going to start, you will have to use all your knowledge you have gained from the first three units.

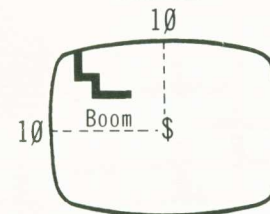
1. The area of the game is limited to that shown in the diagram. As the game starts, the computer scatters 5 mines (*) randomly. 3 seconds later the mines disappear (using CLS).



2. After the mines disappear from the screen (but not from the computers memory) you will see the treasure hunter (■) and the treasure(\$). Your task is to bring the treasure hunter to the treasure without stepping on the invisible mines.



3. The treasure hunter can move right or down, but treading on a mine makes the word 'Boom' appear next to the mine.



Having read the first three units you have gained all the knowledge to write this game program by yourself. Check each stage by running it on the computer. Solve whatever problems you come across.

Note

We recommend that you write the program using DIM. If you find it hard please refer to the following hints.

Hint 1

The main problem of the game is to scatter 5 mines and to keep their positions in memory in such a way that, when the treasure hunter comes across one of them, the computer will display the message 'Boom'.

To do this we have to use 2 arrays:

```
DIM Y(5)  DIM X(5)

Y(1)  ,   X(1)  First mine

Y(2)  ,   X(2)  Second mine

Y(3)  ,   X(3)  Third mine

Y(4)  ,   X(4)  Fourth mine

Y(5)  ,   X(5)  Fifth mine
```

Hint 2



What does the following program do?

```
1Ø DIM Y(3)
2Ø DIM X(3)
3Ø FOR N=1 TO 3
4Ø LET Y(N) = INT (RND*5)
5Ø LET X(N) = INT (RND*5)
6Ø PRINT AT Y(N), X(N); "*"
7Ø NEXT N
```

Hint 3

In order that the computer will print 'Boom' when the treasure hunter comes across a mine use the IF command and do not forget to use the AND (see page 11 in this Unit).

The full program is in the back of the book.

Answer 39

Improvements in the Game

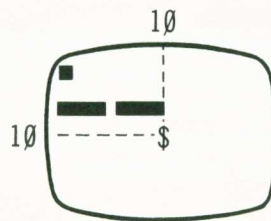
Try to make the following improvements (beware of memory limitations).

- Before beginning of the game, the computer will ask you for your name. When you find the treasure it will print "Congratulations Jeronimo" or any other remark which you may wish displayed.
- If you step on a mine it will display 'Boom' and the game will stop.
- If you find the treasure the computer will show you the mines. This time we will not delete them, in order to show you the danger you managed to pass.

Is the game too easy?

When the mines disappear you will draw a line across this section of the screen with a gap in it.

You can only go through this gap (and don't forget to beware of the mines).



Perhaps the computer will tell you how long it took you to reach the treasure.

The improvements that we have included are on page 50 Answer 40. Now carry on and improve your program yourself.

Answer 40

CONCLUSION

You have now come to the end of the third study Unit. We hope that you now are a more competent programmer and that you can now write programs that may have seemed rather difficult at the beginning.

- You have learnt how to animate characters.
- You have learnt to construct games.
- You have found out how to start improving your programs.

If you are still interested in programming you might like to try Unit 4. Details can be found on the back cover of this book, along with information about other Units in the series.

SUMMARY INDEX

Below is a list of terms that you have studied. Go through them and check if you remember them, (You can refresh your memory by referring to their relevant page numbers which appear in brackets).

INKEY\$	(8)	RND	(31)
IF...THEN	(11)	INT	(32)
INPUT	(22)	OR	(37)
String Variables	(25)	DIM	(37)
Colour	(28)	Array	(38)
AND	(29)		

Answers

Answer 1

We said that when the computer arrives, during the course of the program, at INKEY\$ it reads the key pressed at that moment and carries on. If more than one key is pressed it ignores all but the first and carries on.

Conclusion:

The computer is in a loop all the time from line 10 to line 20. It returns to line 10 all the time, without doing anything. Look at the lower part of the screen, if there is no message it means that the computer is working. If you press BREAK the program will stop and a message will appear at the bottom of the screen.

Answer 2

The computer is going backwards and forwards from line 10 to 20. Each time the computer executes line 10 it prints the key pressed. Therefore the more or longer you press a key, the more it's character will be displayed on the screen.

Answer 3

```
15 BEEP 0.1,10
```

Answer 4

```
10 PRINT AT 10,15:INKEY$  
20 GO TO 10
```

Answer 5

```
15 IF INKEY$="" THEN GO TO
30
16 BEEP 0.2,10
30 FOR Y=21 TO 1 STEP -1
35 PRINT AT Y,15;"*"
37 PAUSE 5
40 PRINT AT Y,15;"□"
50 NEXT Y
```

Line 16 is just to convince you that the computer is RUNNING all the time between lines 10 and 20.

Answer 6

Add the line:

```
60 GO TO 10
```

Answer 7

Nothing will happen: since the condition $2=3$ never exits the computer ignores line 17 and carries on.

Answer 8

If you remember, the command NEW tells the computer to erase the program from memory. Therefore when you press 'Z' the computer executes line 18 and memory is cleared.

Answer 9

```
10 LET N=3
20 PRINT AT N,N;N
30 LET N=N+1
40 IF N>10 THEN GO TO 60
50 GO TO 20
60 PRINT "END"
```

You can see from the length and structure of this program that the program using the FOR...NEXT commands is more sophisticated.

Answer 10

10 PRINT AT 21,15;"\$"		<i>The space ship on the prowl</i>
20 FOR X=0 TO 31		
25 PRINT AT 5,X;">"	}	
29 BEEP 0.1,X		
28 IF INKEY\$="P" THEN GO TO		<i>The aliens are flying across</i>
30 PRINT AT 5,X;"□"		<i>the screen</i>
35 NEXT X		
40 GO TO 20		
50 FOR Y=20 TO 0 STEP-1		
60 PRINT AT Y,15,"*"	}	
63 BEEP 0.1,X		
65 PRINT AT Y,15,"□"		<i>The missile rises</i>
70 NEXT Y		
80 CLS		<i>Clear the screen</i>
90 GO TO 10		<i>Start again with another alien</i>

Note how we have used the loop variables in the BEEP commands on lines 29 and 63. The colours of the spaceships we leave to you.

Answer 11

We shall add the following lines to the previous program:

```
5 LET N=0
10 PRINT AT 0,2;N
85 LET N=N+1
```

Answer 12

You should have chosen the second possibility: when X=15 the aliens could be hit.

```
IF X=15 THEN LET M=M+1
```

(If you have put your own space ship in a different position from ours then change the value of X accordingly.)

Answer 13

The following lines should be put in the program:

```
4 LET M=0
16 PRINT AT 0,30;m
87 IF X=15 THEN LET M=M+1
```

Answer 14

We shall add the line:

```
75 IF N=0 THEN STOP
```

(If you have a memory shortage then reduce the number of the missiles you are able to fire.)

Answer 15

```
75 IF N=10 THEN PRINT AT 0,15;
M/N*100: STOP
```

Note: After the **THEN** more than 1 instruction can be written, each separated by a colon.

The computer will only carry out all these commands if the condition is true

Answer 16

```
10 LET X=15
20 LET Y=10
30 PRINT AT Y,X;0
40 IF INKEY$=CHR$ 9 THEN LET X=X-1 (to the right →)
50 IF INKEY$=CHR$ 11 THEN LET Y=Y-1 (upwards ↑)
60 IF INKEY$=CHR$ 10 THEN LET Y=Y+1 (downwards ↓)
70 IF INKEY$=CHR$ 8 THEN LET X=X-1 (to the left ←)
100 GO TO 30
```

Answer 17

```
25 INPUT A
```

Answer 18

```
80 IF INKEY$="P" THEN GO TO 25
```

Answer 19

```
83 IF INKEY$="" THEN GO TO 30
85 BEEP 0.1,X+Y
```

Answer 20

0.1, -1.4, 3

Answer 21

"4*7=" is a string
4*7 is an arithmetic expression

Answer 22

The following are illegal variable names:

BB\$, C3\$, 4\$

Answer 23

```
27 INPUT "COLOUR?";B
30 PRINT INK B;AT Y,X;A$
```

Pay attention to line number 27. How we can print the "Input question" on the screen.

Answer 24

The following lines stop the pencil moving off the screen.

```
90 IF X>31 THEN LET X=31
92 IF X<0 THEN LET X=0
```

Add two lines to make sure the pencil won't move off the screen from the top or bottom.

Answer 25

We have to multiply by 6:

```
10 PRINT RND*6
```

Answer 26

We have to add 1 and then the next number will be random.

1, 2, 3, 4, 5, 6:


```
10 PRINT INT (RND*6)+1
```

Answer 27

```
10 PRINT INT (RND*3)
20 GO TO 10
```

Answer 28

```
10 PAUSE 0
30 CLS
40 LET M=INT (RND*6)+1
50 LET N=INT (RND*6)+1
60 BEEP 0.2,M*3: PRINT PAPER M
; INK 9;AT 10,10;M
70 BEEP 0.2,N*3: PRINT PAPER N
; INK 9;AT 10,20;N
80 GO TO 10
```

 The computer will not continued unless you press a key.

 The throwing of the dice

 Showing the dice in colour

 Making the sounds.

Answer 29

Add the line:

```
72 IF M=N THEN FOR K=1 TO 7;
  BORDER K: BEEP 0.2,10: BEEP 0.2
  ,20: BEEP 0.2,8: NEXT K
```

Answer 30

```
10 PRINT INK INT (RND*7); AT INT
  (RND*32);"■"
15 BEEP 0.1, INT (RND*20)
20 GO TO 10
```

Answer 31

```
10 PRINT AT 7,10;"XXXXX"
20 INPUT "HOW MANY BOMBS?□□□ ";M
30 FOR N=1 TO M
40 PRINT AT INT (RND*22), INT (
  RND*32);"■"
50 NEXT N
60 PRINT AT 0,0;M
```

Display the submarine
 Ask how many bombs are to be used (M)
 Fire M bombs at random places on the screen
 Print the number of bombs used

We can improve the game and the program more and more, try it yourself!

Answer 32

```
5 LET X1=0
10 LET N=INT (RND*3)+1
20 IF N=1 THEN LET X1=X1+1
30 PRINT AT 3,X1;"1"
40 GO TO 10
```

Answer 33

```

10 LET X1=0
20 LET X2=0
30 LET X3=0
40 LET N=INT (RND*3)+1
50 PRINT AT 3,X1;"1"
60 PRINT AT 6,X2;"2"
70 PRINT AT 9,X3;"3"
80 IF N=1 THEN LET X1=X1+1
90 IF N=2 THEN LET X2=X2+1
100 IF N=3 THEN LET X3=X3+1
110 GO TO 40

```

Initialising the variables

The computer displays the horses

Increase the variable of the horse (that has been randomly chosen) by one.

Go back to the random choosing of a horse

Note: This time we have used the PRINT commands before the IF commands, so that the computer will show the three horses on the starting line before the race begins.

Answer 34

The program lines to draw the finishing line are:

```

2 FOR K=2 TO 10
4 PRINT AT K,20;"■"
6 NEXT K

```

Answer 35

The line to stop the race once a horse has won could be:

```

75 IF X1=20 OR X2=20 OR X3=20
THEN STOP

```

Answer 36

```

34 INPUT "Which horse do you
wish to bet on?";A
36 PRINT A

```

Answer 37

```

10 DIM A(4)
20 FOR N=1 TO 4
30 INPUT A(N)
40 NEXT N
50 FOR N=1 TO 4
60 PRINT "A(";N;")=";A(N)
70 NEXT N

```

The dimensioned array is filled with the values you input

Prints the values of each of the array variables (elements)

Answer 38

A race of six horses:
We shall give just the commands that deal with the race itself.

```

10 DIM X(6)
20 LET N=INT (RND*6)+1
30 LET X(N)=X(N)+1
35 BORDER N
40 PRINT PAPER N;INK 9; AT 3*N ,
X(N); N
50 GO TO 20

```

All the improvement we shall leave to you.

Answer 39

```

5 LET Y=0
7 LET X=0
10 DIM Y(5)
20 DIM X(5)
30 FOR N=1 TO 5
40 LET Y(N)=INT (RND*11)
50 LET X(N)=INT (RND*11)
60 PRINT AT Y(N),X(N);"*"
70 NEXT N

```

Initialisation of the pencil position

Dimensioning the two arrays

Filling the two arrays with random values and printing bombs on the screen.

```

80 PAUSE 100
100 CLS
110 PRINT AT 10,10;"$"
120 PRINT AT Y,X;"■ "
130 IF INKEY$= CHR$ 10 THEN LET
Y=Y+1
140 IF INKEY$= CHR$ 9 THEN LET
X=X+1
150 FOR N=1 TO 5
160 IF Y(N)=Y AND X(N)=X
170 NEXT N
180 GO TO 120

```

A pause
Clear the screen
Displaying the treasure hunter
and the treasure
The pencil
Checking the pencil's position,
does it clash with a mine?

Answer 40

We have added the following improvements

```

3 LET A=0
60 PRINT INK 2; AT Y(N),X(N);"*"
110 PRINT INK 4; AT 10,10;"$"
112 PRINT AT INT (RND*4)+4,0;"■■■■■■■■■■"
122 PRINT AT 0,20;A: LET A=A+1
125 IF INKEY$="" THEN GO TO 122
145 BEEP 0.2,10
160 IF Y(N)=Y AND X(N)=X THEN
PRINT AT Y+1,X;"BOOM": GO TO 200
175 IF Y=10 AND X=10 THEN PRINT
AT 12,20;"CONGRATULATIONS":GO TO 200
200 FOR N=1 TO 5
210 PRINT INK 2; FLASH 1; AT Y(
N),X(N);"*"
220 NEXT N

```

← the clock initialisation
← red mines
← green treasure
← the pathway
← Displaying the clock
and incrementing the clock
Showing the mines

Take note of the FLASH command in line 210:
you will find this command on the 'V' key.

FLASH 1 - Flashing print
FLASH 0 - Stops the flickering.

This series of self-instruction books will teach you the secrets of writing programs in BASIC on your SPECTRUM+ computer.

Unit 1: **FIRST STEPS IN BASIC**

Starting with the first things every programmer needs to know, you will learn to issue commands to the computer, as well as writing and running programs. By the end of the unit you'll be able to make your computer perform useful and interesting tasks.

Unit 2: **EXPLORING BASIC**

This unit teaches you the most important concepts of BASIC: numeric variables, string variables, FOR... NEXT and IF... THEN statements, and much more. You'll create a digital computer clock, and interesting graphics programs including animation.

Unit 3: **COMPUTER GAMES**

In this unit you will learn to develop various computer video games. As you progress through the unit, new programming concepts such as random numbers will be introduced. By the end you'll have considerable programming skills.

Unit 4: **CREATIVE GRAPHICS**

This unit will enable you to continue to expand your programming skills. You'll move step by step through essential mathematical concepts, like sine and cosine, for use in creating amazing on-screen graphics.

Unit 5: **JOURNEY INTO MEMORY**

This unit takes you deep into the computer's memory. You'll learn about how memory is organised, how to design your own characters to replace the standard ones, and how to produce your own graphics symbols.

Unit 6: **ADVANCED PROGRAMMING**

This unit is intended to develop a variety of programs using the knowledge you have accumulated in the first four units. Although bringing together the skills already gained there is still plenty to learn. The important data storage concept of two dimensional arrays is covered, allowing you to develop a large program called Word Game.

And after that . . . we're planning more units to deal with special subjects, such as 3D graphics, machine code and more.



Glentop Publishers Ltd
Standfast House
Bath Place
High Street Barnet
Herts EN5 1ED

£2.95

ISBN 0-907792-62-6



9 780907 792628